have relationships with the following elements: an initial value, occurrence constraint descriptors, value constraint descriptors, and access constraint descriptors, and occurrence constraint descriptors.

[0147] A datatype descriptor describes the type of value(s) allowed to be assigned to a data holder such as an attribute instance or parameter instance. A data descriptor has a zero or one association relationship with an initial value that is a configuration containing the values for each attribute in a model descriptor of the type specified in the datatype descriptor. Any implementation of the datatype descriptor may be used to create an instance of the initial value model. That instance is configured using these values to create the initial value. A datatype descriptor has a zero-to-many aggregation relationship with occurrence constraint descriptors that are descriptions of restrictions to determine how many values can be assigned to the data placeholder implementation. In addition, a datatype descriptor has a zero-to-many aggregation relationship with value constraint descriptors that are descriptions of restrictions to restrict the allowed values held by a placeholder implementation. A datatype descriptor also has a zero-to-many association relationship with access constraint descriptors that describe restrictions to prevent access to the feature implementation unless the user holds certain credentials.

[0148] A datatype descriptor describes a datatype, but does not implement that datatype, hold the data value, or know how to access or change that value. No additional events are added by the datatype descriptor.

[0149] A configuration contains the name of an implementation (model implementation or primitive) and the data instances necessary to set each attribute for that implementation. A configuration is used to construct a new instance of that implementation and "configure" it. The use of a configuration allows implementations to be referenced by name rather than requiring instances of implementations in the modeling tool. A configuration has a one-to-one association relationship with the implementation name that is the name of the implementation of the object this configuration can configure. A configuration has a zero-to-many association relationship with data instances that are the name and value to use to setup or configure an instance. These data instances may be any appropriate classifier instance including a datatype instance or model instance. A configuration describes the instance it configures, but does not implement that instance. A data event is fired whenever a value is accessed, a change is attempted, or a change is applied. A data event is also fired when the implementation name is changed.

[0150] An attribute descriptor of the present invention is a feature descriptor and aggregates a datatype descriptor. Attribute descriptors describe data held by the model. This data may be a primitive data value or a composite data value like an instance of a model. Attribute descriptors also indicate the proper relationship between the attribute and the attribute's owner model. Attribute descriptors have relationships with a datatype descriptor and a relationship type and may have relationships with the following elements: access constraint descriptors, retrieval failure descriptors, storage failure descriptors, prechange signal descriptors and postchange signal descriptors.

[0151] An attribute descriptor has a one-to-one aggregation relationship with datatype descriptor that is the type of value(s) allowed to be assigned to the value of an instance of the attribute. An attribute descriptor has a zero-to-many association relationship with access constraint descriptors that are each a description of restrictions to prevent access to the attribute implementation unless the user holds certain credentials. Access constraint descriptors are inherited from the feature descriptor that an attribute descriptor of the present invention extends. An attribute descriptor has a one-to-one association relationship with a relationship type that is the type of relationship between the attribute owner and the attribute value. Generally the relationship types are association relationship, aggregation, and interaction relationships. An attribute descriptor also has a zero-to-many association relationship with retrieval failure descriptors that are the failures that may occur when an attempting to get the attribute. An attribute descriptor also has a zero-to-many association relationship with storage failure descriptors that are the errors that may occur when an attempting to set the value of the attribute. An attribute descriptor also has a zero-to-many association relationship with prechange signal descriptors that define notification events that occur before a change is applied to an attribute. The prechange signal also defines how to register and deregister interest in receiving these events. An attribute descriptor also has a zero-to-many association relationship with postchange signal descriptors that define notification events that occur after an attribute is changed. The postchange signal also defines how to register and deregister interest in receiving these events.

[0152] An attribute descriptor describes an attribute, but does not implement that attribute, hold the attribute value, or know how to access or change that value. Also no additional events are added by the attribute descriptor.

[0153] An operation descriptor of the present invention is a descriptor describing functionality or actions. Operation descriptors may have relationships with the following elements: parameter descriptors, failure descriptors, suboperation descriptors and access constraint instances.

[0154] Operation descriptors have a zero-to-many aggregation relationship with parameter descriptors that provide details about the data that is be passed into the operation in order for the operation to be performed. Some parameters may affect the way an operation behaves and other parameters hold data upon which to operate. Operation descriptors have a zero-to-many association relationship with failure descriptors that are each a description of the errors that may occur when an operation is attempted. Operation descriptors have a zero-to-many association relationship with suboperation descriptors that are each a list of the operations that this operation descriptor goes through in the order this operation would be executed. These suboperation descriptors are similar to "pseudo code" used to write easy to understand versions or source-code in a programming language. Operation descriptors have a zero-to-many association relationship with access constraint descriptors that are restrictions to prevent access to the feature implementation unless the user holds certain credentials.

[0155] An operation descriptor describes an operation, but does not implement that operation or know how to invoke that operation. No additional events are added by the operation descriptor.

[0156] A parameter descriptor of the present invention is a feature descriptor describing data passed as input or